

---

**<SJTU>**

---

**<足迹微信小程序>**  
**软件架构文档**

**版本 <1.0>**

<项目名称>	Version: <1.0>
软件架构文档	Date: <25/04/22>

修订历史记录

日期	版本	说明	作者
<25/04/2022>	<1.0>	<初步制定软件架构文档>	<刘骏霖，李思旷，张子谦，姜凯>

<项目名称>	Version: <1.0>
软件架构文档	Date: <25/04/22>

目录

1.	简介	4
1.1	目的	4
2.	逻辑视图	5
2.1	概述	5
2.2	在构架方面具有重要意义的设计包	5
3.	部署视图	8
4.	实现视图	9
5.	技术视图	9
6.	数据视图（可选）	错误!未定义书签。
7.	核心算法设计（可选）	错误!未定义书签。
8.	质量属性的设计	9

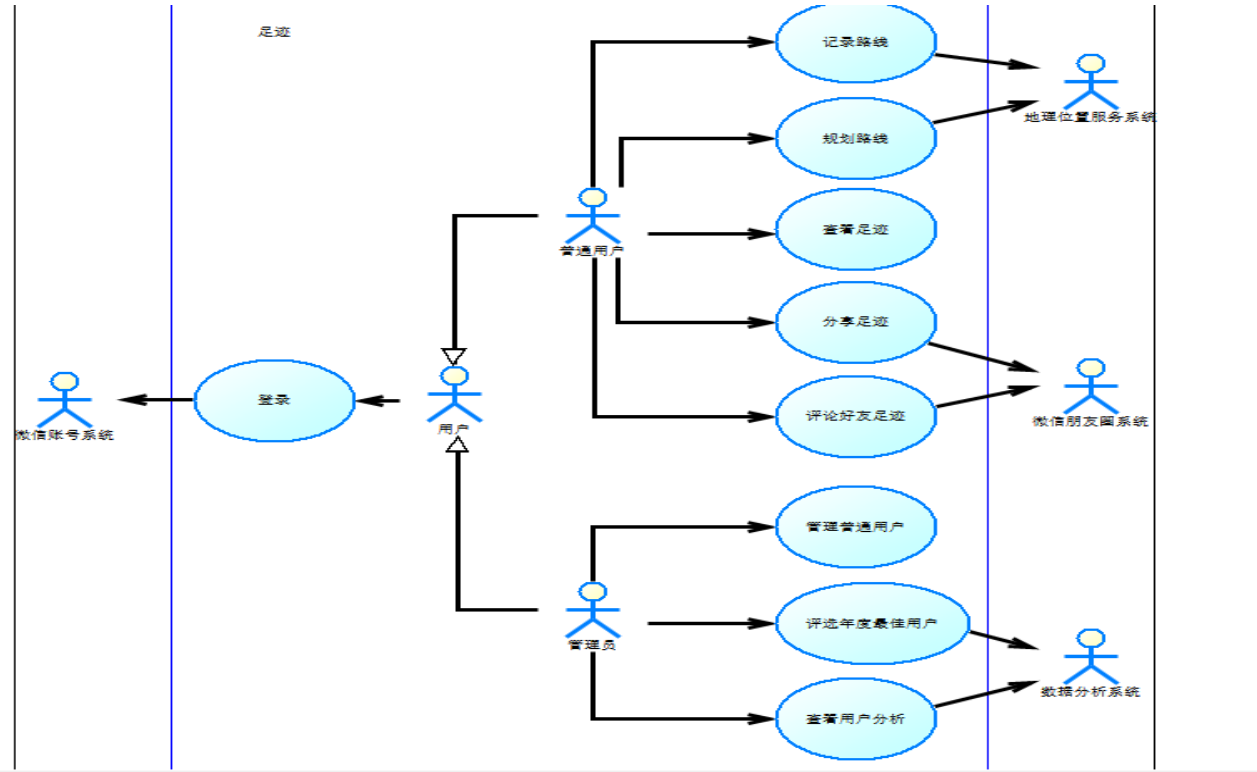
<项目名称>	Version: <1.0>
软件架构文档	Date: <25/04/22>

# 软件架构文档

## 1. 简介

### 1.1 目的

本文档的目的是为“足迹”微信小程序提供软件架构层面的概括与介绍，包括各类视图与相关算法的设计介绍。



用户分为普通用户和管理员两类，统一通过微信账号系统登录，根据用户身份的不同可以有不同的用例情况，具体如下：

普通用户：

- 1、记录路线：允许普通用户记录行进的路线。
- 2、规划路线：通过在地图上以标注的形式规划相应的路线。
- 3、查看足迹：查看在某个时间下在某个地区的具体信息（通过照片视频等方式呈现）。
- 4、分享足迹：通过微信朋友圈系统分享足迹。
- 5、评论好友足迹：可以对好友用户的足迹发表评论。

管理员：

- 1、管理普通用户：管理普通用户的账号信息等。
- 2、评选年度最佳用户：允许管理员评选出足迹年度最佳用户，并推送到其他用户的足迹软件中进行展示。
- 3、查看用户分析：允许管理员查看用户分析，包括用户数量、上传照片、路线信息的统计分析。

<项目名称>	Version: <1.0>
软件架构文档	Date: <25/04/22>

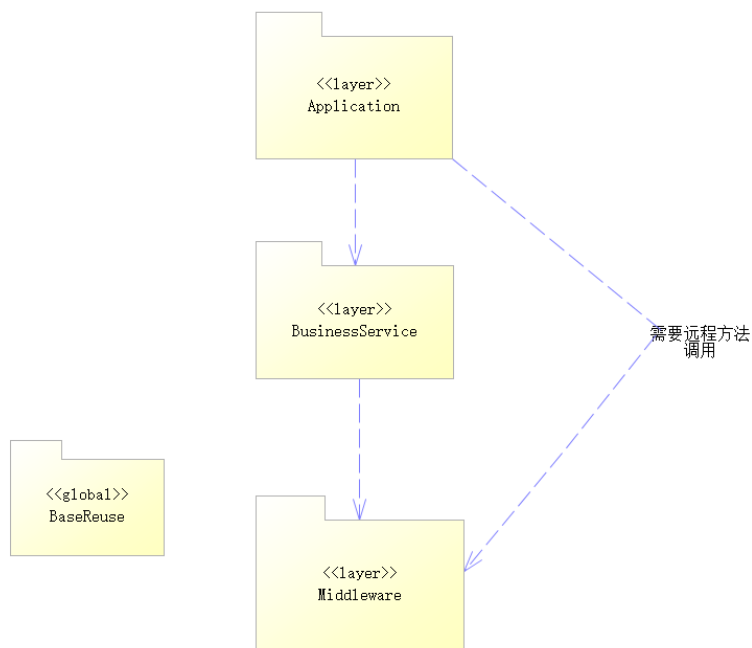
## 2. 逻辑视图

### 2.1 概述

逻辑架构:

Application 依赖于 BusinessService 依赖于 Middleware, Application 同时依赖于 Middleware 的远程调用

选择分层架构模式

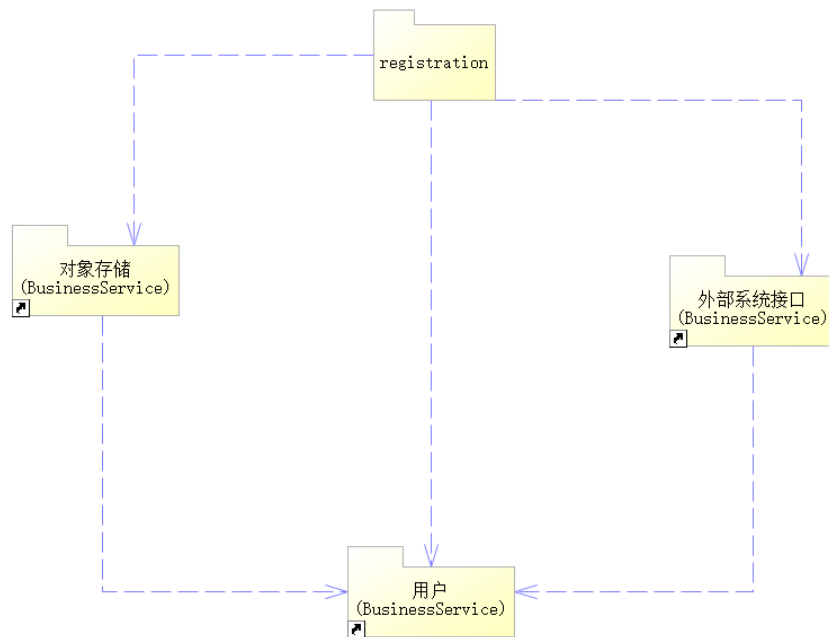


### 2.2 在构架方面具有重要意义的设计包

1、Application:

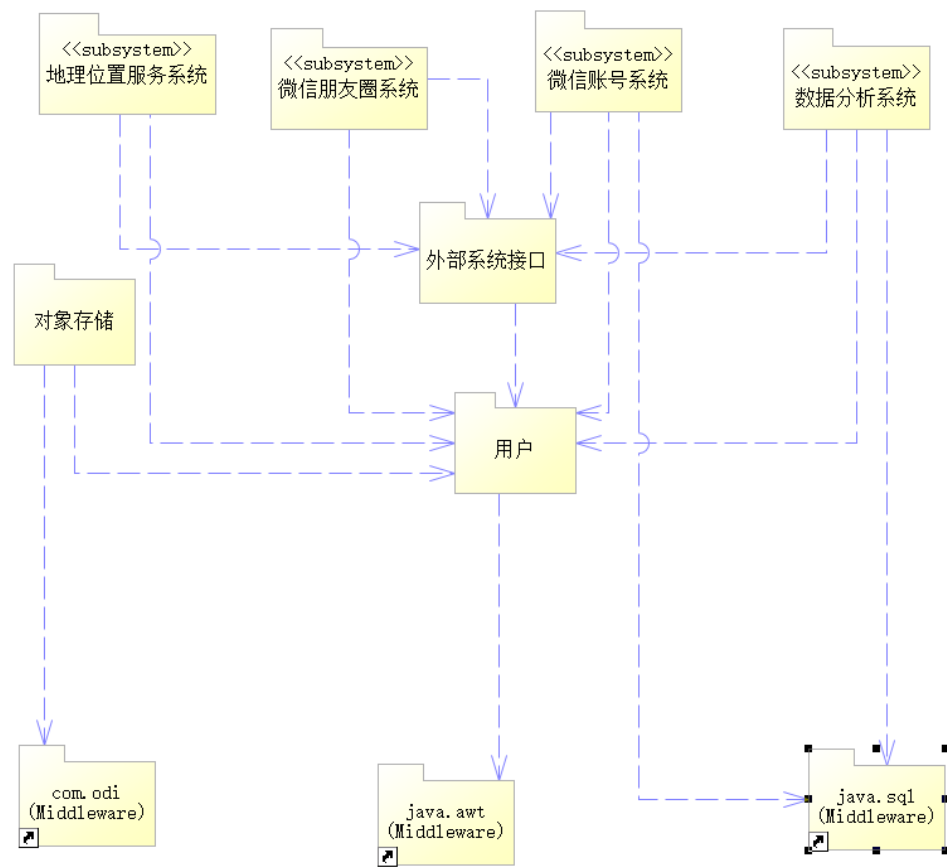
Registration 依赖于对象存储和外部系统接口

<项目名称>	Version: <1.0>
软件架构文档	Date: <25/04/22>

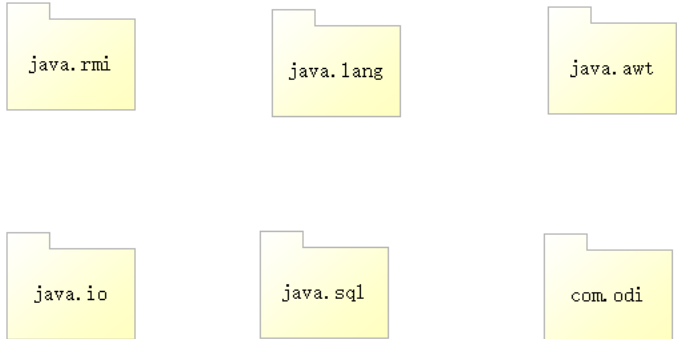


2、Business Service:

<项目名称>	Version: <1.0>
软件架构文档	Date: <25/04/22>



3、Middleware:

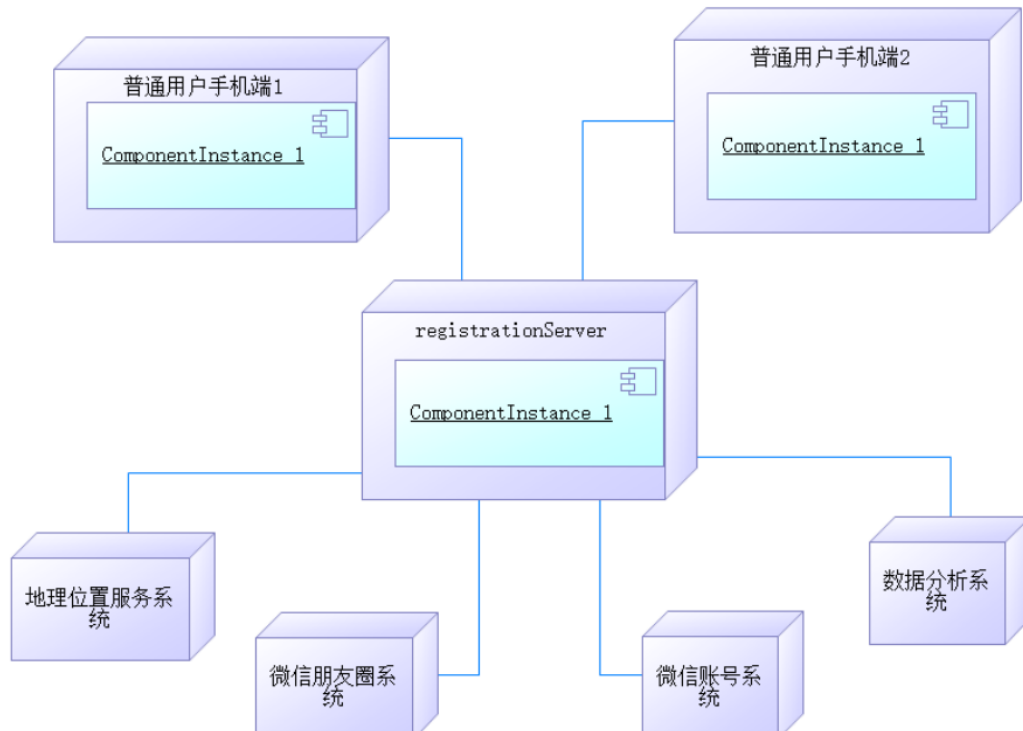


<项目名称>	Version: <1.0>
软件架构文档	Date: <25/04/22>

### 3. 部署视图

物理架构：

手机端请求服务将会与各个服务系统相连接

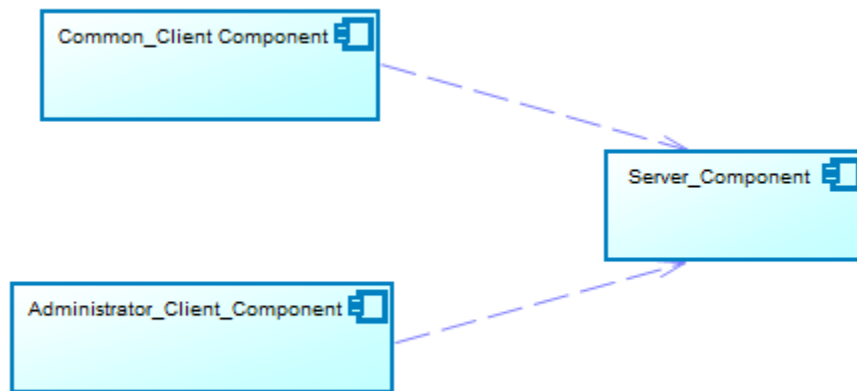


注册服务端与普通用户手机端连接，接受用户请求，同时与地理位置服务系统，微信朋友圈系统，微信账号系统，数据分析系统连接，以为用户提供服务。



<项目名称>	Version: <1.0>
软件架构文档	Date: <25/04/22>

#### 4. 实现视图



普通用户与管理员用户分别与服务端相连，向服务端发送请求并由服务端提供相应服务。

#### 5. 技术视图

编程语言: JavaScript, Java

开发工具: 微信开发者工具

框架: Taro, Spring MVC

数据库: MySQL

#### 6. 质量属性的设计

软件架构明确了整个系统的结构组成与详细分工，其用例视图明确了系统面向用户的核心功能，使得研发过程中对功能的开发更具有针对性；逻辑视图的设计将系统分解为从问题域中得到的一系列关键抽象，即对象、类和包，这些抽象为系统提供了功能分析，在本次设计中采用分层架构模式进行设计，将不同层次的功能部分抽象出来并分别加以详细说明，从而使得整个架构更加清晰，更具有可扩展性；部署视图的设计则从物理层面明确了本次开发中系统的详细结构，保证在将软件系统部署到不同环境时不必大量修改源代码，提高了可移植性。